Course – Java and Advanced Java Programming Concepts
Duration: 300 Hours
Prerequisites: Knowledge of computer fundamentals
(Note: Each Session is of 5-6 hours)

Session 1: Getting Started
Lecture:
- Setup development environment (JRE, JDK, eclipse)
- Features of java
- JVM Architecture
- JDK and its usage
- Structure of java class
- Writing your first Java program
- About main () method
- Constructor in Java

Lab:
Write Java programs to:
- Print Hello World
- Add two numbers/binary numbers/characters
- Calculate compound interest
- Calculate power of a number
- Swap two numbers

Session 2: Introduction to Class and Objects
Lecture:
- Class & Object
- Access Specifier
- Java Data Types, Primitives and Binary Literals

Lab:
Write Java programs to:
- Calculate area of rectangle
- Calculate area and circumference of circle using multiple classes
- Java program to find ASCII value of a character

Session 3: Operators
Lecture:
- Arithmetic Operator
- Relational Operator
- Logical Operator
- Unary Operator
- Ternary Operator
- Assignment Operator

Session 4: Conditional and Looping Statements
Lecture:
- If, else if, switch
- break & continue keyword
- for loop

- while loop
- do while loop
- static & final keyword
- Recursion

Lab:

Write Java programs to:

- Display prime numbers between 1 and 100 or 1 and n
- Swap two variables without using the third variable
- Find the factorial of a number
- Check if a number is palindrome or not
- Print Fibonacci series till n
- Add two integer variables in 5 different ways using functions and control statement
- Find square root of a number without sqrt method
- Check Armstrong number
- Calculate grades of students using their marks
- Use switch case, recursion, print patterns, etc.

Session 5: Arrays

Lecture:

- Initializing an Array in Java
- Two dimensional array in java
- Java Variable Arguments explained
- Add, update, read array elements
- Sorting and searching in array
- Java String Array to String
- How to copy arrays in Java

Lab:

Write Java programs to:

- Calculate average of numbers using Array
- Reverse an array
- Sort an array in ascending order
- Convert char Array to String
- Add two Matrix using Multi-dimensional Arrays
- Sort strings in alphabetical order
- Find out the highest and second highest numbers in an array
- Concatenate two arrays

Sessions 6 & 7: Object Oriented Programming

Lecture:

Introduction to OOP concepts
Encapsulation
Inheritance: single & multilevel
Inheritance: Hierarchical
Polymorphism: Compile time and runtime polymorphism
Rules of overriding and overloading of methods
super and this keywords
Upcasting & downcasting of a reference variable

Lab:

Create a class Employee and encapsulate the data members.

Create demo applications to illustrate different types of inheritance.

Session 8: Abstract Class and Abstract Methods
Lecture:
Abstract class and abstract methods
Interface (implementing multiple interfaces)
Final variables, final methods and final class
Functional interface
New interface features(Java 8 & above)
Lambda expression and stream API
Arrays
Enumerations
Lab:
Create an Array of Employee class and initialize array elements with different employee objects.
Try to understand the no of objects on heap memory when any array is created.

Session 9: Access Modifiers and Garbage Collection
Lecture:
Access modifiers(public, private, protected and default)
Packages and import statements
Static imports
Constructor chaining (with and without packages)
Accessing protected variables and methods outside the package
Garbage collection in java
Requesting JVM to run garbage collection
Different ways to make object eligible for garbage collection: (Nulling a reference variable, Re-assigning a reference variable & island of isolation)
Finalize method
Lab:
Create a demo application to understand the role of access modifiers.
Implement multilevel inheritance using different packages.
Access/invoke protected members/methods of a class outside the package.
Override finalize method to understand the behavior of JVM garbage collector.

Sessions 10 & 11: Wrapper Classes and String Class
Lecture:
Wrapper classes and constant pools
String class, StringBuffer & StringBuilder class
String pool
Lab:
Create sample classes to understand boxing & unboxing.
Use different methods of java defined wrapper classes.
Create StringDemo class and perform different string manipulation methods.
Tutorial:
Understand the difference between String / StringBuffer / StringBuilder.

Sessions 12 & 13: Exception Handling
Lecture:

Exception hierarchy, Errors, Checked and un-checked exceptions
Exception propagation
try-catch-finally block , throws clause and throw keyword
Multi catch block
Creating user defined checked and unchecked exceptions
Lab:
Create user defined checked and unchecked exceptions .

Sessions 14 & 15: java.io, java.nio and java.utils Package
Lecture:
Brief introduction to InputStream, OutputStream, Reader and Writer interfaces
NIO package
Serialization and de-serialization
Shallow copy and deep copy
Object Class & java.util Package
Date, DateTime, Calendar class
Converting Date to String and String to Date using SimpleDateFormat class
Object Class: Overriding to String, equals & hashcode method
Lab:
Create a Demo class to Read & write image/text files.
Create SerializationDemo class to illustrate serialization and de-serialization process.

Sessions 16 & 17 & 18: Collections
Lecture:
Introduction to collections: Collection hierarchy
List, Queue, Set and Map Collections
List Collection:
- ArrayList, LinkedList
- Vector (insert, delete,search,sort, iterate, replace operations)

Collections class
Comparable and Comparator interfaces
Queue collection
Lab:
Create DateManipulator class to convert String to date, date to String and to find out number of days between two dates.
Create a List of java defined wrapper classes and perform insert/delete/search/iterate/sort operations.
Create a collection of Employee class and sort objects using comparable and comparator interfaces.
Implement Queue data structure using LinkedList and Queue collection.
Lecture:
Set Collection:
- HashSet, LinkedHashSet & TreeSet collection
- Backed set collections

Map Collection:
- HashTable, HashMap, LinkedHashMap & TreeMap classes
- Backed Map collections

Generics
Concurrent collections

Lab:
Create an Employee HashSet collection and override equals & hashCode methods to understand how the set maintains uniqueness using these methods.
Create a Sample class to understand generic assignments using "? extends SomeClass" , "? super
someclass " and "?".

Sessions 19 & 20: Multithreading & Synchronization
Lecture:
MultiThreading : Thread class and Runnable Interface
sleep, join, yield, setPriority, getPriority methods
ThreadGroup class
Lab:
Invoke private methods of some other class using reflection.
Create multiple threads using Thread class and Runnable interfaces.
Assign same task and different task to multiple threads.
Understand sleep, join, yield methods.
Lecture:
Synchronization
Deadlock
Wait, notify and notifyAll methods
Inner classes
Lab:
Create a Deadlock class to demonstrate deadlock in multithreading environment.
Implement wait, notify and notifyAll methods.
Demonstrate how to share threadlocal data between multiple threads.
Create multiple threads using anonymous inner classes.

Sessions 21, 22 & 23: Database & SQL
Lecture:
Introduction to Relational Model
Understanding Basic SQL Syntax
SELECT, INSERT, UPDATE, DELETE
Querying Data with the SELECT Statement
The SELECT List
SELECT List Wildcard (*)
The FROM Clause
How to Constrain the Result Set
DISTINCT and NOT DISTINCT
• Filtering Results with the Where Clause
WHERE Clause
Boolean Operators
The AND Keyword
The OR Keyword
Other Boolean Operators BETWEEN, LIKE, IN, IS, IS NOT
• Shaping Results with ORDER BY and GROUP BY
ORDER BY
Set Functions
Set Function And Qualifiers

GROUP BY
HAVING clause
• Matching Different Data Tables with JOINs
CROSS JOIN
INNER JOIN
OUTER JOINs
LEFT OUTER JOIN
RIGHT OUTER JOIN
FULL OUTER JOIN
SELF JOIN
• Creating Database Tables
CREATE DATABASE
CREATE TABLE
NULL Values
PRIMARY KEY
CONSTRAINT
ALTER TABLE
DROP TABLE

Sessions 24 & 25
Lecture:
J2EE Overview
• J2EE Container
• Packaging Web applications
• J2EE compliant web application
• Deployment tools.
• Web application life cycle
• Deploying web applications.
• Web Services Support
JDBC & Transaction Management
• Introduction to JDBC API
• JDBC Architecture
• JDBC Drivers
• JDBC Classes & Interfaces: Driver, Connection, Statement, PreparedStatement, ResultSet
• Stored procedures and functions Invocation
• Design Pattern: Data Access Object Pattern
Lab :
• Perform database CRUD operations using JDBC classes and interfaces.

Session 26: Architecture of Web
Lecture:
Brief history of the Internet
How does the Internet work?
Internet Protocol; HTTP
Domain Names; Domain Name Service servers
HTTP Protocols
o Difference between HTTP1.0, HTTP 1.1, and HTTP 2.0
o Methods – GET, POST, HEAD, PUT, DELETE, etc.
o Status codes

o Stateless nature of the protocol and HTTP Session
o HTTPS
Architecture of the Web
Web servers – IIS, Apache server
Lab:
Exploring different browsers
o Mozilla Firefox, Google Chrome, Safari
Exploring different text editors
o Windows: Notepad++, Linux: Gedit or Vim or Emacs

Sessions 27 & 28: HTML
Lecture:
Introduction to HTML
Document Object Model (DOM)
Basic HTML Tags
o Alignment, Headings, Anchor, Paragraph, Image, Lists, Tables, and iFrames
HTML5
o New features in HTML5
o New elements, new attributes, link relations, microdata, ARIA accessibility
o HTML5 Validation
o Audio & Video Support
HTML Forms & Controls
o Input, Text Area, Radio Button, Checkbox, Dropdown, Submit, Reset, Button, etc.
Lab:
Create a HTML form for building a resume.

Sessions 29 & 30: Cascading Style Sheets (CSS)
Lecture:
Introduction to CSS, Styling HTML with CSS, Structuring pages with CSS,
Inline CSS, Internal CSS, External CSS, Multiple styles, CSS Fonts
CSS Box Model
id Attribute, class Attribute
HTML Style Tags
Linking a style to an HTML document
Lab:
Apply inline, internal and external CSS to change colors of certain text portions, bold, underline, and italics certain words in the previously created HTML resume form.

Session 31: Responsive Web Design
Lecture:
Introduction of UI Scripting
The Best Experience for All Users
o Desktop, Tablet, Mobile
Bootstrap
o Overview of Bootstrap, Need to use Bootstrap
o Bootstrap Grid System, Grid Classes, Basic Structure of a Bootstrap Grid
o Typography
o Components – Tables, Images, Jumbotron, Wells, Alerts, Buttons, Button Groups, Badges/Labels, Progress Bars, Pagination, List Groups, Panels, Dropdowns, Collapse,

Tabs/Pills, Navbar
o Forms, Inputs
o Bootstrap Themes, Templates
Lab:
Update the design of the Resume form using Bootstrap

Sessions 32 & 33: JavaScript
Lecture:
Introduction to JavaScript
Variables in JavaScript
Statements, Operators, Comments, Expressions, and Control Structures
JavaScript Scopes
Strings, String Methods
Numbers, Number Methods
Boolean Values
Dates, Date Formats, Date Methods
Arrays, Array Methods
Lab:
Practice writing basic JavaScript programs for better understanding of the language constructs
Lecture:
Objects, Object Definitions, Object Properties, Object Methods, Object Prototypes
Functions, Function Definitions, Function Parameters, Function Invocation, Function Closures
Object Oriented Programming
o Method, Constructor, Inheritance, Encapsulation, Abstraction, Polymorphism
Lab:
Write a JavaScript program to sort a list of elements by implementing a sorting algorithm.
Write a JavaScript program to list the properties of a JavaScript object.

Sessions 34 & 35: JavaScript DOM
Lecture:
Document Object Model (DOM)
o Object hierarchy in JavaScript
o HTML DOM, DOM Elements, DOM Events
o DOM Methods, DOM Manipulation
Forms, Forms API, Forms Validation
Regular Expressions
Errors, Debugging
Introduction to Browser Dev Tool
Pushing code quality via JSLint tool
Lab:
Write a JavaScript function to get First and Last name from the previously created Resume form
Validate the entire Resume form using client-side JavaScript
Write a JavaScript function to validate whether a given value is RegEx or not.

Sessions 36, 37 & 38:
Lecture:

• Hibernate Framework
o Introduction to Hibernate Framework
o Architecture
• Hibernate in IDE
o Creating web application using Hibernate API
o Lifecycle of Hibernate Entities
• HB with annotation example
• Hibernate Mappings and Relationships
• Collection and Component Mapping
• HQL, NamedQueries, Criteria Queries
Lab:
• Demonstrate Hibernate as standalone library in Java application
• Develop a web application (Online Bookshop) using Hibernate Persistence
Reading: Study Hibernate architecture from www.hibernate.org/docs

Sessions 39, 40 & 41:
Lecture:
• What is Spring Framework
• Overview of Spring Architecture
• Spring Modules Overview
• Understanding Spring 4 annotations(Basic Introduction)
• What is IoC (Inversion of Control)
• IOC container
• Dependency Injection
• Spring Beans and its lifecycle
• Autowiring Beans
• Configuring collections
• Spring Validations
• Spring i18n, Localization, Properties
• File Upload example
Lab:
• Design and deploy Library Management System using Spring

Session 42 & 43:
Lecture:
• Spring Boot essentials
• Why Spring boot
• Spring Boot Overview
• Basic Introduction of MAVEN
• Building Spring application with Boot
• Spring Boot in detail (Use Spring Boot for all demo & assignments here onwards)
• Running a web application using Spring Boot with CRUD (with Static Data not DB)
Lab:
• Create Hello World Spring Boot Web application
• Check Libraries imported by Spring Boot
• Create Spring Boot CRUD application

Sessions 44 & 45:
Lecture:

Spring Data Module
• Spring Data JPA (Repository support for JPA)
• CrudRepository&JPARepository
• Query methods
• Using custom query (@Query)
Lab:
• Add CRUD operations with Spring JPA etc. to earlier Spring Web application.

Sessions 46 & 47:
Lecture:
Building REST services with Spring
• Introduction to web services
• SOAP Vs RESTful web services
• RESTful web service introduction
• Create RESTful web service in java using Spring Boot
• RESTful web service JSONexample
• RESTful web service CRUD example
• Using POSTMAN client to invoke REST API's
• REST service invocation using REST Template
Lab:
• Create REST API for Employee Management using Spring Boot
• Invoke it from POSTMAN app
• Invoke it from another Spring Boot Web application using REST Template

Sessions 48 & 49:
Lecture + Lab: Unit Testing
• Introduction to unit testing
• Introduction to Junit
    - Fix the annotations
    - Assert Exceptions
    - Run Tests
• Introduction to Mockito
    - Create DAO and BO Layer
    - Adding Mockito Dependency
    - Stubbing and Setting Expectation
    - Result verification
• Unit Testing of Spring Service Layer
• Integration Testing of Spring Applications: REST API

Session 50: ES6 & Typescript
- Var, Let and Const keyword
- Arrow functions, default arguments
- Template Strings, String methods
- Object de-structuring
- Spread and Rest operator
- Typescript Fundamentals
- Types & type assertions, Creating custom object types, function types
- Typescript OOPS - Classes, Interfaces, Constructor, et